IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 October 2024, accepted 6 December 2024, date of publication 11 December 2024, date of current version 19 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3514673

RESEARCH ARTICLE

2D Instance-Guided Pseudo-LiDAR Point Cloud for Monocular 3D Object Detection

RUI GAO[1], JUNOH KIM[2], AND KYUNGEUN CHO[3], (Member, IEEE)
[1]Department of Multimedia Engineering, Dongguk University, Seoul 04620, Republic of Korea
[2]NUI/NUX Platform Research Center, Dongguk University, Seoul 04620, Republic of Korea
[3]Division of AI Software Convergence, Dongguk University, Seoul 04620, Republic of Korea

Corresponding author: Kyungeun Cho (cke@dongguk.edu)

This work was supported in part by the Korea Institute of Police Technology (KIPoT) Grant funded by the Korean Government [Korean National Police Agency (KNPA)] (AI Driving Ability Test Standardization and Evaluation Process Development) under Grant 092021D75000000 (50%), in part by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government [Ministry of Science and ICT (MSIT)] under Grant 2022R1A2C2006864 (40%), and in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) under the Artificial Intelligence Convergence Innovation Human Resources Development Grant funded by the Korean Government (MSIT) under Grant IITP-2024-RS-2023-00254592 (5%), and in part by the MSIT (Ministry of Science and ICT), Korea, under the ICAN (ICT Challenge and Advanced Network of HRD) support program (RS-2023-00260248) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation) (5%).

ABSTRACT Monocular three-dimensional (3D) scene understanding tasks, e.g., object size angle and 3D position, estimation are challenging to perform. More successful current methods usually require data from 3D sensors, which limits their performance if only monocular images are used because of the lack of distance information. In recent years, 3D object detection methods based on pseudo-LiDAR have effectively improved the accuracy of 3D prediction. However, most pseudo-LiDAR methods directly use LiDAR-based 3D detection networks, which also limits their performance. In this paper, a new monocular 3d object detection framework is proposed. By redesigning the representation of pseudo-LiDAR, a 2d detection mask channel is introduced as a guide layer to guide 3D object detection. A new 3D object detection network is designed for the newly designed multi-channel data. The pillar feature encoding module is optimized by using 2d detection mask and height histogram, so that it can more effectively convert point clouds into pillar features by using 2d detection mask and height characteristics of different objects. A new feature extraction network was designed using the transformer module to extract pillar features. The transformer head was optimized using three independent heads to categorize the position, properties and specific parameters of the object. An evaluation on the challenging KITTI dataset demonstrates that the proposed method significantly improves the performance of state-of-the-art monocular methods.

INDEX TERMS Autonomous driving, deep learning, monocular 3D object detection.

I. INTRODUCTION

Fields such as autonomous driving and robot navigation require access to the three-dimensional (3D) data of objects for a clear understanding of the environment. This requires the use of perceptual systems to obtain precise environmental information accurately. 3D object detection is one of the most important tasks for understanding 3D physical world phenomena. Existing 3D object detection methods can be broadly categorized into two groups depending on whether the input data are 2D or 3D sensor data [1]. Light Detection and Ranging (LiDAR) sensors are effective for distance measurements and are more resistant to bad weather than cameras. However, LiDAR data are unstructured and sparse, which makes LiDAR processing more challenging. Although LiDAR-based methods have exhibited promising performance, expensive and bulky sensors limit their application. In addition, LiDAR scans virtual objects that do not exist for certain highly reflective materials. This has a significant impact on perceiving the presence of objects in the environment [2], [3]. Although LiDAR can provide accurate depth information, its data density is still very low compared to cameras, especially for long-distance objects and low-cost LiDAR sensors. And the camera can capture rich

The associate editor coordinating the review of this manuscript and approving it for publication was Chuan Li.

color, texture and detail information, especially for distant objects. The rapid development of image-based 2d detection in recent years also shows that image information is very useful in object detection. Methods based on inexpensive and easily deployable cameras have exhibited significant potential in many scenarios. Therefore, despite the greater challenges of estimating 3D bounding boxes from images, this task continues to attract considerable attention and is gradually becoming a popular topic in the computer vision community.

3D object detection remains challenging because of the lack of 3D data if only image information is used. Various deep-learning-based methods have been developed to solve this problem. Typical image-based 3D object detection methods [4], [5], [6], [7] use a process similar to that of a 2D detector, focusing mainly on the RGB features that are extracted from the 2D image to predict the 3D data of the object. Owing to the lack of spatial information, these methods are unsuitable for 3D object detection-related tasks. This is one of the main reasons that early studies failed to achieve superior performance [8]. The pseudo-LiDAR-based approach, which is a pioneering concept that was first introduced in [9], aims to utilize image features to mimic LiDAR point cloud data while using depth estimation networks and LiDAR-based 3D object detection algorithms [10]. Pseudo-LiDAR enhances the reliability of the image-based depth estimation by mimicking high-quality LiDAR signals. Pseudo-LiDAR achieved an accuracy of 45.3% in the KITTI vehicle detection task, which is more than three times better than that of traditional monocular or stereo 3D algorithms [10]. In this approach, in which monocular images are converted into LiDAR representations, both pseudo-LiDAR and variations in the coordinate system are important for 3D object detection [11], [12].

In order to make effective use of the camera's features of capturing rich color, texture and detail information, this study first uses the depth estimation algorithm to convert images into point clouds, and designs a new 3D object detection network that can integrate 2D detection results and pseudo-LiDAR point clouds as input. Compared with the 3D object detection network that only uses LiDAR data as input, In this study, more dense and rich detailed data and color features collected by cameras can be effectively used. The performance of 3D estimation can be improved more effectively. Point cloud data explicitly show spatial information, and the network can learn richer features because several specific spatial structures exist only in 3D space. A previous study [12] noted that the focus of improving 3D object detection is the transformation of the coordinate system, rather than the data representation. In order to improve the performance of 3d object detection more effectively. In this study, pseudo-LiDAR based 3d object detection was used to make better use of spatial information to compensate for the lack of depth data in image information. Most of the existing 3d object detection algorithms based on pseudo-LiDAR are designed directly using LiDAR-based algorithms. Since the pseudo-LiDAR point cloud density is much higher than the LiDAR point cloud data, most algorithms directly perform down-sampling processing on pseudo-LiDAR. While this can effectively utilize the spatial features of pseudo-LiDAR, direct down-sampling can't take advantage of the image's data-intensive data details. In order to break through this limitation, better use of rich and dense image features. In this study, a new 3d object detection framework is proposed. A new multi-channel dense point cloud is formed by fusing 2d detection mask with pseudo-LiDAR point cloud, and then a new 3d object detection network designed for multi-channel dense point cloud is used to obtain 3d object detection results with higher quality. Specifically, we first perform monocular depth estimation to obtain the depth data for each image pixel and convert it into a point cloud in terms of [x, y, z] to represent the 3D data. We design a multimodal feature fusion module to add the object information that is obtained from the 2D bounding box to the 3D point cloud data in the form of confidence to use the 2D bounding box extracted based on RGB image features and the 3D spatial features that can be efficiently utilized simultaneously. That is, we add a confidence level for each point in the point cloud, which indicates the probability that the point is an object. The final expression for the data is [x, y, z, confidence]. Subsequently, based on our data representation, a novel 3D object detection network is proposed. For multi-channel point cloud data, the pillar feature encoding (PFE) module is re-designed to convert the input point cloud into pillars with regular spatial arrangement while filtering and differentiating between the object and the background. After that, the spatial attention is introduced into the feature extraction network to amplify and suppress the feature data of the pseudo-images of the point cloud and enhance the feature extraction capability of the feature extraction network. Extensive experiments on the KITTI dataset validate the merits of the proposed approach. Notably, our results on the 3D object detection benchmark are significantly better than those of existing monocular methods.

In summary, the contributions of this study are four fold:

- An efficient monocular 3D object detection system based on pseudo-LiDAR representation is proposed. The novel 3D object detector improves the detection performance by introducing 2D object detection based on 2D image features into 3D point clouds and considering the inherent noise and inaccuracy of the pseudo-LiDAR representation.
- We design a multimodal feature fusion module for merging the 2D bounding box and 3D point cloud extracted based on RGB image features to utilize the 2D detection results and 3D spatial features effectively.
- A novel 3D object detection network is designed. Compared with traditional Pillar-based networks, our network effectively utilizes 2D mask and spatial attention to improve performance.

- The results show that our approach yields the best performance on the KITTI Bird's Eye View (BEV) and 3D benchmarks, significantly outperforming state-of-the-art methods.

The remainder of this paper is organized as follows: Section II outlines previous research, presents a detailed comparison of existing 3D detection algorithms, and provides a summary and analysis of LiDAR-based, image-based, and pseudo-LiDAR-based methods. Each section ends with a summary of the limitations of previous approaches and the improvements offered by this study. Section III explains the proposed method, explaining the methodological details of each module of the proposed framework. Section IV presents a performance comparison of the proposed framework with mainstream single-objective 3D detection algorithms through experiments. In addition, ablation experiments to analyze the effectiveness of each module of the proposed framework are described. Finally, Section V concludes the paper and discusses the advantages and limitations of the method. The future application directions and improvement directions of this method are also outlined.

II. RELATED WORKS

A. LiDAR-BASED 3D OBJECT DETECTION

LiDAR is an active depth measurement sensor that scans the surrounding environment with thousands of pulses to obtain distance information. LiDAR-based 3D detection algorithms are favored by researchers and used extensively owing to their reliability. Although our approach is designed for monocular image data, we perform 3D object detection based on point clouds, which is similar to LiDAR-based methods. Therefore, we present several typical LiDAR-based methods. Existing methods can be broadly categorized into point-based, grid-based, and hybrid representations [13].

PointRCNN [14] is a point-based detector that extracts features from the raw LiDAR point cloud via PointNet++ [15] and sends the feature proposals to Region of Interest pooling pooling refinement for further detection. However, the computational cost of point-based methods is extremely high; thus, they are not suitable for processing large-scale point clouds during autonomous driving. In contrast, grid-based methods discretize the point cloud into a structured grid such that 2D or 3D convolutions can be applied. VoxelNet [16] converts a 3D point cloud into equally spaced voxels, aggregates the point features within each voxel, and encodes the voxel features using dense 3D convolution. SECOND [17] improves the efficiency by introducing a novel sparse 3D convolution for LiDAR point clouds. PointPillars [18] further improves the efficiency of the network by organizing the point cloud into pillars instead of voxels to encode point cloud features using 2D convolution. Some studies have argued that grid-based methods lead to the loss of fine-grained information. Therefore, hybrid methods have been proposed to merge point features into grid representations [19], [20].

Although all LiDAR-based point cloud methods achieve good detection results owing to the accuracy of LiDAR sensor data, pseudo-LiDAR point clouds are much denser than LiDAR sensor scanned point cloud data. However, the estimated depth cannot be exactly the same as the reality owing to the performance limitations of depth estimation networks and the existence of numerous textureless regions as well as various occlusions in RGB images, which result in substantial uncertainty in the depth estimation. Therefore, the point cloud data obtained from our pseudo-LiDAR are relatively dense, complex, and noisy. We believe that the characteristics of pseudo-LiDAR point clouds cannot be effectively utilized if LiDAR-based methods are applied directly. Therefore, to narrow the gap with LiDAR-based algorithms further, we design a novel 3D object detection network for the characteristics of pseudo-LiDAR point clouds. The selection of the point cloud coding method is crucial owing to the high density and large amount of noise in the pseudo-LiDAR point cloud. The PointPillars strategy [18], which involves randomly extracting effective pillars from the dense information region, is efficient and can effectively constrain the regularization of the 3D object detection network so that it is not limited by the point cloud data. Considering the inherent noise and inaccuracy characteristics of pseudo-LiDAR representations, the 3D object detection network designed in this study is based on PointPillars and a self-attention module.

B. IMAGE-BASED 3D OBJECT DETECTION

As opposed to LiDAR-based methods that require accurate 3D point cloud data, monocular methods require only a single image, given both an RGB image and the corresponding camera parameters. The goal of image-based 3D object detection is to classify and localize objects of interest, which makes the 3D object detection task more challenging.

The aim of monocular image-based 3D object detection is to predict 3D information regarding an object from a single image using well-established 2D object detection frameworks and geometric priors. These approaches can generally be categorized into two-stage methods and one-stage anchored and unanchored tandem-based methods. Although the detection accuracy of such algorithms is not high, they are favored in industry and academia owing to their low cost and high potential research value [10]. Most older monocular 3D object detection algorithms employ 2D object detection as a base model and extend it to 3D object detection using different techniques (e.g., regression) [21], [22]. Mono3D [5], [23] uses a priori knowledge from monocular images (e.g., the object size and ground plane) to generate 3D object proposals. Deep3DBox [24] introduces a discretized representation with perspective constraints based on the fact that the 3D bounding box should closely fit the 2D detection bounding box. Deep MANTA [4] uses key points to encode 3D vehicle information because they are rigid objects with well-known geometry. Methods such as MonoDLE [25], PGD [26], and PackNet [27] integrate

multi-scale feature fusion and attention mechanisms for depth map estimation and error analysis, thereby improving the performance. MonoDETR [28] employs adaptive feature aggregation via depth-guided converters for monocular 3D object detection.

Image-based 3D object detection methods usually rely on RGB features that are extracted from 2D images. Some methods can enhance the understanding of a 3D scene through depth map prediction. However, owing to the lack of spatial information, RGB features extracted from 2D images are not suitable for 3D-related tasks [8]. LiDAR data describe real-world 3D coordinates in the form of point clouds, which are significantly more explicitly represented. Therefore, the pseudo-LiDAR method, which converts the estimated depth information into a point cloud representation to simulate the real LiDAR data for better handling of 3D detection tasks, has been proposed to enhance the usefulness of depth information. Pseudo-LiDAR enables image-based methods to achieve a significant performance improvement in 3D tasks by providing a clearer spatial representation. Therefore, a pseudo-LiDAR-based method forms the basis of this study.

C. PSEUDO-LiDAR-BASED 3D OBJECT DETECTION

With the development of depth estimation and LiDAR-based 3D object detection algorithms, a new process known as pseudo-LiDAR has been proposed [8], [9], [29], [30], [31], [32], [33]. The goal of this process is to build a bridge between image-based and LiDAR-based methods. In this pipeline, dense depth estimation is first performed from the image, converting each pixel of the image into depth data. Subsequently, all pixels are projected into 3D space to generate pseudo-LiDAR point cloud data. Thereafter, the pseudo-LiDAR point cloud data are directly used as input to apply the LiDAR-based detection method. The study [9] was one of the earliest works to use the pseudo-LiDAR process. These authors believed that generating pseudo-LiDAR point clouds can better utilize the spatial characteristics of point clouds to improve the 3D object prediction performance. On this basis, ForeSeE [29] was designed as a foreground–background separation monocular depth estimation method, which effectively improves the 3D object detection performance. DA-3Ddetp [30] improves the performance by adapting features of an unsound image-based pseudo-LiDAR domain to an accurate real LiDAR domain using domain adaptation techniques. AM3D [8] and Mono3D_PiDAR [33] use 2D detection masks to extract point cloud frustums from pseudo-LiDAR data and then generate 3D bounding boxes for each frustum. This method effectively guides 3D object detection using 2D masks, significantly improving the detection performance by narrowing the search space for potential 3D objects. The studies [8], [9], [29], [30], [31], [32], [33] all directly used LiDAR-based 3D detection algorithms for 3D object detection after converting images into pseudo-LiDAR data. The main distinction between these works lies in how they improved performance, either by optimizing the point cloud data or refining the training methods for pseudo-LiDAR.

Common LiDAR sensors that are used in cars usually scan the surrounding environment with 64 or 128 rotating beams to generate sparse point cloud data, while optical cameras can operate at high frame rates and provide dense depth maps. Therefore, the data density of pseudo-LiDAR is substantially greater than the point cloud data scanned by LiDAR sensors. This study presents a new 3D object detection network for dense pseudo-LiDAR point clouds to make better use of this feature, thereby optimizing the 3D detection performance. Considering that previous studies [8], [33] used 2D detection masks to improve the object detection performance, this study also introduces 2D detection masks as the basis for the foreground–background separation technology. However, as opposed to previous methods, this study does not directly use 2D detection masks to extract point cloud frustums from pseudo-LiDAR point clouds. Instead, the 2D detection masks are used as the confidence of each point in the point cloud to distinguish the foreground from the background for guiding the generation of pseudo-LiDAR point clouds. The 3D object detection network is redesigned on this basis. The network is based on the PointPillar architecture [18], and a new point cloud coding module is proposed for the high density but low accuracy of pseudo-LIDAR point clouds. In addition, this study introduces a Transformer and convolutional neural network (CNN) [34], [35] as new feature extraction modules. In this manner, the feature extraction capability of the 3D detection network in processing pseudo-LiDAR point clouds is significantly enhanced, thereby effectively improving the 3D object detection performance.

D. MONOCULAR DEPTH ESTIMATION AND 2D OBJECT DETECTION

Although monocular depth estimation and object detection are not the main focus of this study, their performance has a significant impact on the overall effectiveness of the proposed method. The performance of monocular depth estimation algorithms based on deep learning has been improved significantly with the rapid development of deep learning technology. These algorithms have made great strides in terms of accuracy and efficiency, providing more reliable and accurate depth estimation capabilities for various application scenarios. Depth Anything [36] uses zero-shot transfer learning by employing a data engine and introducing auxiliary supervision to improve the depth estimation performance, robustness, and generalizability of the model. In addition, because several pseudo-LIDAR-based algorithms [9], [15], [31] perform depth estimation using DORN [37], we also used the depth estimation results from DORN for the ablation experiments in this study. With the development of deep convolutional neural networks (CNNs), traditional methods that rely on manually designed feature extraction have gradually been eliminated and CNN-based deep-learning algorithms have become the mainstream in 2D object detection. At present, popular 2D detection networks far outperform 3D

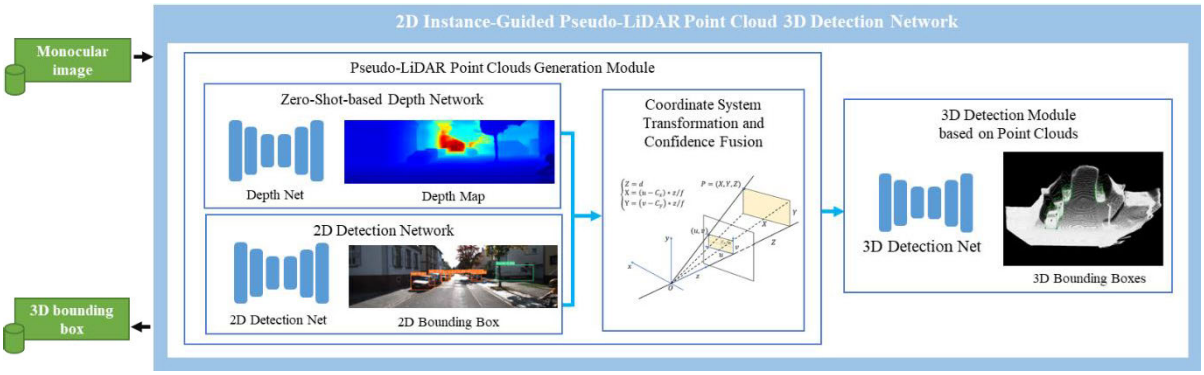

FIGURE 1. Overview of proposed method.

object detection networks Therefore, we selected the easily deployable YOLOv8 [38] as the 2D object detection module in this study.

III. PROPOSED METHOD

The aim of this study was to estimate the 3D bounding box of an object in an image using a single monocular image. The proposed approach does not rely on any additional data sources, such as LiDAR, depth cameras, or stereo images during training and testing, and only assumes that the camera matrix is known. Similar to other methods [33], [39], this approach parameterizes the output of the 3D bounding box into seven parameters, including the 3D coordinates of the object center (x, y, z), dimensions of the object (h, w, l), and heading angle θ of the object. This approach aims to achieve accurate localization and measurement of objects in 3D space without the need for complex sensors, thereby reducing the cost and complexity and enhancing the feasibility of practical applications.

Pseudo-LiDAR-based methods reduce the performance gap with LiDAR and outperform other image-based methods in 3D detection owing to advantages such as the ability to utilize the spatial properties of point clouds. Therefore, we selected the pseudo-LiDAR-based method as the foundation for our framework. Most pseudo-LiDAR-based methods directly utilize LiDAR-based 3D detection algorithms for 3D object detection, and we believe that pseudo-LiDAR data have significant variability from the point cloud data scanned by LiDAR sensors, representing a difference in density but high inaccuracy. Therefore, using LiDAR-based 3D detection algorithms directly for 3D object detection cannot effectively exploit the characteristics of pseudo-LiDAR data. We believe that this limits the performance of existing pseudo-LiDAR-based methods. Therefore, we redesign the 3D detection network and introduce a novel depth estimation network to optimize the generation of pseudo-LiDAR point clouds and to improve the 3D detection performance further. Considering that 2D object detection outperforms 3D object detection, a 2D detection network is introduced for further enhancement. As shown in Figure 1, our proposed method consists of two main parts: a pseudo-LiDAR point cloud generation module with 2D instance guidance and a 3D object detection network based on the pseudo-LiDAR point cloud. The 3D object detection network is trained using a novel pseudo-LiDAR point cloud with 2D instance data. A multi-task loss function, including the classification, regression, and angle and direction classification losses, is employed to optimize the 3D bounding box estimation. The pseudo-LiDAR point cloud generation module can be updated with the latest monocular depth estimation and 2D object detection algorithms at any time. This allows the framework to be highly flexible and to maintain cutting-edge performance owing to the increasing advances in such algorithms. The redesigned 3D object detection network consists of three parts: the pillar feature encoder, feature extraction module, and center head. We develop a novel adaptive weight pillar feature encoder, which retains more foreground points while mitigating the impact of excessive background points in 3D detection, to address the challenges posed by dense pseudo-LiDAR data. In addition, we design a new feature extraction network that combines the global features extracted by a Transformer with the local features obtained by a CNN. Finally, we utilize three independent prediction heads to achieve the output of the final parameters. We analyze the impact of each module on the overall performance in detail in the experimental section.

A. 2D INSTANCE-GUIDED PSEUDO-LiDAR POINT CLOUD GENERATION

The foundational step of the monocular 3D target detection process is to estimate the depth of a monocular image. Therefore, the pre-trained Depth Anything model is used directly for depth estimation. The weights of the depth estimation network are not updated during training; instead, it is used as an offline module. We also selected other depth estimation networks for the ablation experiments to demonstrate that the process is independent of the selection of the monocular depth estimation network.

In this study, the major focus is on how to use depth information, rather than how to obtain it. The camera calibration

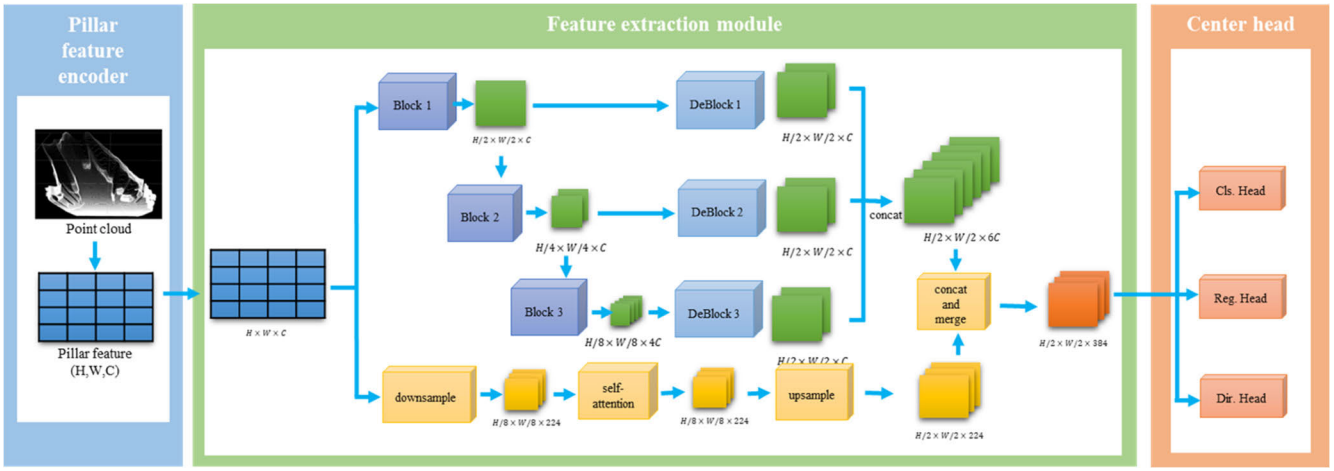

FIGURE 2. Overview of 3D object detection network.

file converts the estimated depth into a point cloud, which is subsequently used as the data input. Specifically, the pixel coordinates (u, v) in the 2D image space with depth $d$can be converted into 3D coordinates in the camera coordinate system (x, y, z) using (1):

$$\begin{cases} z = d \\ x = (u - C_x) * z/f \\ y = \left(v - C_y\right) * z/f \end{cases} \tag{1}$$

where (C_x, C_y) is the principal point and f is the focal length of the camera. Given the camera extrinsic matrix M_E,

$$M_E = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}, \tag{2}$$

where M_E is a 4×4 matrix, R is a 3×3 orthogonal unit matrix, which is also known as a rotation matrix, and t is a 3D translation vector. Subsequently, a pseudo-LiDAR point cloud in the world coordinate system can be obtained as follows:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = M_E^{-1} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}, \tag{3}$$

After the above 2D to 3D data representation conversion, LiDAR-based detection algorithms can be applied to process the pseudo-LiDAR point cloud. Owing to the high density but low accuracy of pseudo-LiDAR point clouds, especially in the detection of distant objects, the point cloud offset is more obvious, which affects the 3D detection accuracy. This study optimizes the detection effect by enhancing the difference between the foreground and background to alleviate this problem. 2D images are rich in appearance features such as texture, color, and shape and the images contain a large amount of visual information, which enables the 2D detector to locate the bounding box of an object accurately. Although the resolution of distant objects is low, 2D images usually retain sufficient semantic information to detect objects accurately. Previous studies [8], [33] have utilized 2D detection masks to extract frustums from pseudo-LiDAR point clouds and generate the corresponding 3D bounding boxes to enhance the 3D object detection. However, relying solely on 2D detection masks for frustum extraction can introduce inaccuracies owing to the limitations in 2D mask precision and errors during the 2D-to-3D conversion, thereby potentially affecting the 3D object detection accuracy. Thus, in this study, instead of directly extracting frustums, we use 2D detection masks to guide the generation of pseudo-LiDAR point clouds. The 2D mask is applied as a confidence score for each point in the cloud to distinguish the foreground from the background. By incorporating confidence values across the entire point cloud instead of focusing on frustum extraction, this approach aims to provide a more reliable foundation for 3D object detection, reducing the impact of errors in the 2D-to-3D transformation. Specifically, a 2D detection mask is combined with a pseudo-LiDAR point cloud, the scores in the 2D detection mask are used to identify foreground areas, and these scores are added to the point cloud as confidence. Points outside the mask are assigned a confidence level of 0 and serve as the background. With the 2D detection mask, the 3D detection network can focus on specific regions, effectively reduce background noise and irrelevant regions in the point cloud, and reduce the computational complexity. In addition, the 2D detection results are based on image information, which can help the 3D network to integrate multi-view and multi-scale information in complex scenes, enhance the network understanding of the 3D geometric features of objects, and improve the 3D object detection accuracy. Our 2D detector is based on Mask2Former [40]. The implementation is as follows: The fraction of each pixel in the predicted bounding box is projected into 3D space, as shown in Equation (1). Thereafter, this score is encoded as the fourth channel of the pseudo-LiDAR point cloud,

as shown in Equation (4).

$$p_c = \begin{bmatrix} x \\ y \\ z \\ conf \end{bmatrix}, \tag{4}$$

where p_c is the pseudo-LiDAR point cloud and $conf$ is the confidence of each pixel of the predicted 2D bounding box.

B. 3D OBJECT DETECTION NETWORK

We design a 3D object detection network based on self-attention module for Pseudo-LiDAR data with 2D detection results. As shown in Figure 2, the main components are (1) a pillar feature encoding module, (2) a feature extraction module, and (3) a 3D box detection head.

1) PILLAR FEATURE ENCODING MODULE

The pseudo-LiDAR point cloud must first be converted into a pseudo-image, following which features are extracted based on the pseudo-image using a 2D convolutional and self-attention architecture. In the pillar-based 3D detection method, the main purpose of the pillar feature encoding (PFE) module is to convert the input point cloud (in this study, the shape of the point cloud is $[x, y, z, conf]$) into pillar, and the pillar have a regular spatial arrangement order, and the feature size of each pillar is fixed. And the results of PFE directly affect the feature extraction module and the final detection accuracy. The current PFE module in 3D object detection has a problem in that it is difficult to distinguish between foreground and background in the scene. This includes too much background data in the encoding process, which affects the accuracy of object detection.

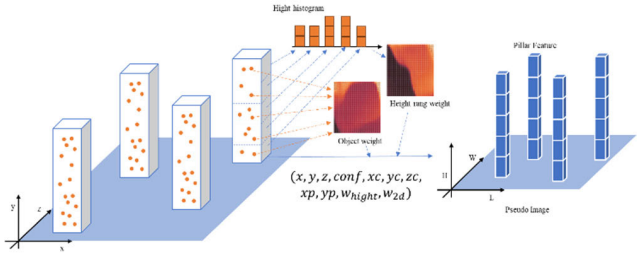

FIGURE 3. Overview of pillar feature encoding module.

To solve this problem, in this study, we propose a novel adaptive weight pillar feature encoder. Based on the original PFE, we add two weight layers, the height range weight layer and the object weight layer. For each point in a pillar, 2 weights are computed by utilizing a height histogram 2D mask to distinguish the foreground and background in the scene. Specifically for all points in each pillar, histogram operations are performed on their height values to divide the height values into regions, each corresponding to a specific height range. Next the points in each height range are counted to obtain the number of points located at a particular height. Since the height of the ground is usually low and the number of points is high, the density of points near the bottom of this pillar is high. Meanwhile, the 2D mask can effectively distinguish the background and foreground, so the range selected by the 2D mask is used as the object weight layer. The two weight layers generated by this method will be used in the pillar feature data generated along with the pillar data for the feature extraction network, this method can retain the semantic and geometric information derived from the height dimension, thus enhancing the understanding of the point cloud data.

Let p represent a point in the pseudo-LiDAR point cloud, with coordinates x, y, z and confidence $conf$. The point cloud is discretized into a uniformly spaced grid in the x-y plane to create a set of pillar sets P. Subsequently, each point in each pillar set is encoded into a 11-dimensional vector D: $(x, y, z, conf, xc, yc, zc, xp, yp, w_{hight}, w_{2d})$, where xc, yc, zc denote the distances to the arithmetic mean points of all points in the grid, $conf$ denotes the confidence of each pixel of the predicted 2D bounding box, xp, yp denote the offsets of the points to the x,y center of the grid, and w_{hight}, w_{2d} denote weight of height range and weight of 2d mask. A limit is imposed on the number of non-empty pillars per sample (P) and number of points in each pillar (N) to create a tensor of size (D, P, N). If excessive data exist in the pillars, they are randomly sampled; if too little data are available, they are filled with zeros. Thereafter, a pseudo-image with dimensions (C, H, W) is created, where H and W denote the height and width of the image, respectively.

2) FEATURE EXTRACTION MODULE

We use a region proposal network (RPN) similar to VoxelNet [16] as the backbone, the structure of which is shown in Figure 2. This part consists of a 2D CNN, which is used to extract high-dimensional features from the pseudo-images output by the first part of the network. The RPN backbone is divided into two sub-networks. One top-down sub-network is used to extract features from feature maps with increasingly smaller spatial resolutions. The other sub-network is responsible for upsampling the features that are extracted from feature maps of different resolutions to the same dimensional size using an inverse convolution operation and then concatenating them. The parameters of the convolutional layer are listed in Table 1.

$Conv2D(C_{in}, C_{out}, k, s, p)$ represents a 2D convolution operator, where C_{in} and C_{out} are the numbers of input and output channels, respectively; and k, s, p are the kernel size, step size and padding size, respectively. Each convolutional layer is followed by the application of BatchNorm and ReLU. The final output feature is a high-resolution feature map that is constructed by concatenating all features at different steps.

As the pseudo-LiDAR point cloud suffers from noise problems, such as displacement and deformation, and the 2D convolution uses a fixed-field-of-view convolution kernel, the global features cannot be extracted efficiently. The enhancement of global features is required to recognize the position and shape of the objects satisfactorily. Therefore, we add a

TABLE 1. Parameters of the convolutional feature extraction module.

Network	Properties	Output dimensions
Input	Encoded point cloud input feature	$H \times W \times C$
Block 1	$Conv2D(64,64,3,2) \times 1$ $Conv2D(64,64,3,1,1) \times 1$	$\frac{H}{2} \times \frac{W}{2} \times 2C$
Block 2	$Conv2D(64,128,3,2) \times 1$ $Conv2D(128,128,3,1,1) \times 5$	$\frac{H}{4} \times \frac{W}{4} \times 2C$
Block 3	$Conv2D(128,256,3,2) \times 1$ $Conv2D(256,256,3,1,1) \times 5$	$\frac{H}{8} \times \frac{W}{8} \times 2C$
DeBlock 1	$ConvTranspose2D(64,128,1,1) \times 1$	$\frac{H}{2} \times \frac{W}{2} \times 2C$
DeBlock 2	$ConvTranspose2D(128,128,2,2) \times 1$	$\frac{H}{2} \times \frac{W}{2} \times 2C$
DeBlock 3	$ConvTranspose2D(256,128,4,4) \times 1$	$\frac{H}{2} \times \frac{W}{2} \times 2C$
Concat	Concat DeBlock 1, DeBlock 2, DeBlock3	$\frac{H}{2} \times \frac{W}{2} \times 2C$

TABLE 2. Parameters of the self-attention-based global feature extraction module.

Network	Properties	Output dimensions
Input	Spatial features of the point cloud	$H \times W \times 64$
Downsample	$Conv2D(64,128,3,2,1) \times 1$ $Conv2D(128,224,3,2,0) \times 1$ $Conv2D(224,224,1,2,0) \times 1$	$\frac{H}{8} \times \frac{W}{8} \times 224$
Self-attention		$\frac{H}{8} \times \frac{W}{8} \times 224$
Upsample	$Conv2D(128,256,3,2) \times 2$	$\frac{H}{2} \times \frac{W}{2} \times 224$
Concat	Concat self-attention-based global feature extraction module and convolutional feature extraction module	$\frac{H}{2} \times \frac{W}{2} \times 608$
Merge	$Conv2D(608,384,3,1,1) \times 1$ $Conv2D(384,384,3,1,1) \times 1$	$\frac{H}{2} \times \frac{W}{2} \times 384$

self-attention mechanism to the feature extraction module to enhance the extraction of global features.

The self-attention module is described in detail below. The encoded point cloud input feature is transformed and flattened into three vectors $Q \epsilon \mathbb{R}^{C_Q \times N}$, $K \epsilon \mathbb{R}^{C_K \times N}$, and $V \epsilon \mathbb{R}^{C_V \times N}$, where C_Q, C_K, and C_V represent the channels of the respective vectors and N is the number of spatial positions. The spatial context of a feature point is obtained from its weighted sum and all other feature points, as follows:

$$O_i = \sum_{j=1}^{N} \omega_{i,j} \psi \left(x_j\right), \tag{5}$$

where O_i is the output context at position i and $\omega_{i,j}$ is the normalized attention weight. ψ indicates a fully connected layer that transforms x (the flattened feature map) into value space. The weight map $W \epsilon \mathbb{R}^{N \times N}$ is computed as

$$W = softmax(\frac{Q^T \times K}{\sqrt{C_K}}), \tag{6}$$

where Q^T is the transpose of the query matrix Q of shape $N \times C_Q$, resulting in a matrix of shape $N \times C_Q$, K is the key matrix of shape $C_K \times N$, $\sqrt{C_K}$ is a scaling factor, and *softmax* is applied to the scaled dot product.

As shown in Figure 2, the first encoded point cloud input feature is fed into the *Downsample* network consisting of three *Conv2D* layers, each of which is followed by BatchNorm and ReLU layers. The features after the *Downsample* network are fed into the self-attention layer, and the features extracted after the self-attention layer are fed into the Upsample network consisting of two *ConvTranspose2D* layers.

The convolutional layer parameters are listed in Table 2. Finally, the global features extracted by the self-attention module are connected to the local features extracted by the CNN. After passing through the Merge module consisting of two Conv2D layers, the final spatial features of the point cloud are obtained.

3) 3D BOX DETECTION HEAD

The design of classification and localization as a unified module is often considered paradoxical in 3D bounding box regression. This is because a unified module will suffer from conflicting objective functions between two different tasks, where shared parameters may interfere with the sensitivity of each individual task [41], [42]. More accurate detection performance can be achieved by regressing the 3D position and attributes of the object separately from the detailed parameters. Therefore, we referred to [43], [44], and [45] to design isolated branches to change the detection head.

TABLE 3. Parameters of 3d box detection head.

Network	Properties	Output dimensions
Input	Encoded point cloud input feature	$H \times W \times 384$
$conv_{cls}$	$Conv2D(384,2,1,1,1) \times 1$	$\frac{H}{2} \times \frac{W}{2} \times 2$
$conv_{box}$	$Conv2D(384,14,1,1,1) \times 1$	$\frac{H}{2} \times \frac{W}{2} \times 4$
$conv_{dir\_cls}$	$Conv2D(384,4,1,1,1) \times 1$	$\frac{H}{2} \times \frac{W}{2} \times 4$

The architecture of the proposed head is shown in Figure 2, and the specific parameters are listed in Table 3. Three branches with dedicated convolutional layers are used for different task domains.

C. TRAINING LOSS

We use the same training functions as those in SECOND [17]. The ground truth and our estimated 3D bounding box can be expressed as $(x, y, z, w, l, h, \theta)$. The results of the ground truth and estimation are computed using Equation (7).

$$\Delta x = \frac{x^{gt} - x^{pred}}{d^{pred}},$$
$$\Delta y = \frac{y^{gt} - y^{pred}}{d^{pred}},$$
$$\Delta z = \frac{z^{gt} - z^{pred}}{h^{pred}},$$

$$\Delta w = log\frac{w^{gt}}{w^{pred}},$$
$$\Delta l = log\frac{l^{gt}}{l^{pred}},$$
$$\Delta h = log\frac{h^{gt}}{h^{pred}},$$
$$\Delta\theta = sin(\theta^{gt} - \theta^{pred}), \quad (7)$$

where Δx is the normalized difference in the x-coordinates of the ground truth x^{gt} and predicted x^{pred} bounding boxes, scaled by d^{pred}; Δy is the normalized difference in the y-coordinates; Δz is the normalized difference in the z-coordinates; Δw is the logarithmic difference in width; Δl is the logarithmic difference in length; Δh is the logarithmic difference in height; $\Delta\theta$ is the sine of the difference in orientation angles; x^{gt} and x^{pred} are the ground truth and predicted boxes, respectively; and $d^{pred} = \sqrt{(w^{pred})^2 + (l^{pred})^2}$.

The total localization loss is:

$$\mathcal{L}_{loc} = \sum_{b\in(x,y,z,w,l,h,\theta)} SmoothL1(\Delta b), \quad (8)$$

where Δb represents the difference for each bounding box parameter b, as defined previously. In addition, $SmoothL1(\Delta b)$ is the Smooth L1 loss, which is a loss function that combines properties of both the L1 and L2 losses.

The focal loss is used for the object classification loss:

$$\mathcal{L}_{cls} = -\alpha_{pred}(1 - p^{pred})^{\gamma} logp^{pred}, \quad (9)$$

where p^{pred} denotes the probability of the predicted object class. α_{pred} is a weighting factor that balances the importance of positive/negative classes. γ is a focusing parameter that adjusts the rate. We use the settings $\alpha = 0.25$ and $\gamma = 2$. Therefore, the total loss is

$$\mathcal{L}_{total} = \beta_{loc}\mathcal{L}_{loc} + \beta_{cls}\mathcal{L}_{cls}, \quad (10)$$

where $\beta_{loc} = 2$ and $\beta_{cls} = 1$.

IV. EXPERIMENTS

This section describes the experiments that were conducted, including the dataset, evaluation criteria, performance comparisons, and ablation experiments.

A. DATASET

The KITTI 3D object detection benchmark dataset [46], which consists of 7,481 training samples and 7,518 test samples, was used in our experiments. It should be noted that our proposed method uses only monocular RGB image data and does not include any LiDAR point clouds or stereo image data during training and testing. Previous methods [8], [33], [47] were followed to divide the original training samples into a training set containing 3,712 samples and a validation set containing 3,769 samples. The data were preprocessed by resizing all images to a resolution of 1242 × 375 pixels. During training, the data were shuffled to avoid order bias.

B. EVALUATION METRICS

The standard evaluation toolkit provided by KITTI, which computes the precision-recall curves and average precision (AP) with Intersection over Union (IoU) thresholds of 0.5 and 0.7, was used in our experiments. The performance of the proposed method was demonstrated for three different levels of difficulty: easy, moderate, and hard. The APs for the BEV and 3D object detection in the experiments were denoted as AP_{BEV} and AP_{3D}, respectively.

C. IMPLEMENTATION DETAILS

The proposed method was developed using PyTorch [48] and OpenPCDet [49]. We restricted the point clouds to a detection range of $[-40, 40]$ m along the x-axis, $[0, 70.4]$ m along the y-axis, and $[-1, 3]$ m along the z-axis. The pillar grid size was set to 0.162 m. As pseudo-LiDAR points are generated by projecting pixels from the predicted depth map into 3D space, the maximum number of points per pillar was set to 128, which is four times the number used for LiDAR points. This allowed the model to capture the denser spatial information present in the pseudo-LiDAR data effectively. The model was trained for 50 epochs with a batch size of 2, using a learning rate of 2×10^{-2} that was decayed by 0.8 every 10 epochs. The training process required approximately 9 h on an NVIDIA RTX 3090 GPU.

D. COMPARISON METHODS AND RESULTS

We used the car category of the KITTI validation set for the performance comparison and analysis. The performance on the validation set was computed using the AP settings of the 40 recall positions for comparison with existing state-of-the-art methods.

As the goal of this study was monocular 3D object detection, we selected methods that also utilize monocular images as input for comparison. Specifically, six advanced methods were selected: MonoDTR [50], MonoDETR [28], NeurOCS-MLC [51], MonoCD [52], Pseudo-LiDAR [9], Mono3D_PiDAR [33], and AM3D [8]. We used the same dataset (KITTI) and applied consistent evaluation metrics across all methods to ensure a fair comparison. This guaranteed that the performance differences were owing to the methods themselves, rather than variations in the data or evaluation protocols. By comparing our approach with these recognized methods, we demonstrated its competitiveness with the current state of the art. Pseudo-LiDAR [9], Mono3D_PiDAR [33], and AM3D [8] are representative pseudo-LiDAR-based methods. Pseudo-LiDAR [9] is the first method to incorporate this process owing to its theoretical significance and application value in 3D object detection. Mono3D_PiDAR and AM3D are based on Pseudo-LiDAR and further utilize 2D detection masks to extract frustums from point clouds for 3D object detection, which reflects the research idea of 3D detection based on 2D information. Our proposed method is also based on this concept, but innovatively uses the score of the 2D detection mask as the

TABLE 4. Comparison of our model with state-of-the-art methods on the car category of the KITTI validation set.

Method		APBEV/AP3D (%), IoU = 0.5			APBEV/AP3D (%), IoU = 0.7		
Name	Reference	Easy	Moderate	Hard	Easy	Moderate	Hard
MonoDTR	CVPR 2022	69.0/64.0	52.4/47.3	45.9/42.2	33.3/24.5	25.3/18.5	21.6/15.5
MonoDETR	ICCV 2023	-	-	-	32.0/23.6	21.4/15.9	17.8/12.9
NeurOCS-MLC	CVPR 2023	-	-	-	39.1/31.2	26.7/21.0	23.6/17.7
MonoCD	CVPR 2024	-	-	-	34.6/26.5	25.0/19.4	21.5/16.4
Pseudo-LiDAR	CVPR 2019	70.8/66.3	49.4/42.3	42.7/38.5	40.6/28.2	26.3/18.5	22.9/16.4
Mono3D_PLiDAR	ICCV 2019	72.1/48.4	53.1/48.3	44.6/43.0	41.9/31.5	28.3/21.0	24.5/17.5
AM3D	ICCV 2019	72.6/68.9	51.8/49.1	44.2/42.2	43.7/32.2	28.3/21.0	23.8/17.2
Ours		**75.8/71.1**	**56.2/52.1**	**52.0/47.1**	**44.7/34.7**	**31.8/25.0**	**27.9/21.2**

confidence level of each point in the point cloud to distinguish the foreground and background more effectively. In addition, a novel 3D detection network is designed for the newly generated point cloud data. Therefore, these methods were selected for comparison to demonstrate the innovativeness of our approach as well as to verify the superiority of its performance under the same research direction. We also selected MonoDTR [50], MonoDETR [28], NeurOCS-MLC [51], MonoCD [52] as representatives of the latest research based on monocular detection to evaluate the performance of these cutting-edge techniques on the same dataset.

The relevant results are shown in Table 4, from which it is clear that our proposed method outperformed existing methods at all difficulty levels. Our method exhibited average enhancements of 7 and 7.7 in BEV and 3D with IoU $= 0.5$ compared to the most primitive Pseudo-LiDAR method, which uses only images as inputs and a deep prediction network for conversion into a pseudo-LiDAR point cloud, and then applies a 3D detection algorithm for object detection. With IoU $= 0.7$, the BEV and 3D enhancements were 4.9 and 5.9, respectively, on average. Improvements of 9.8 and 8.6 in the accuracy of the 3D bounding box were achieved for the moderate and hard levels with an IoU of 0.5. This indicates that the proposed method exhibits a significant performance improvement in the detection of more distant objects.

In addition, our proposed method significantly outperformed both other pseudo-LiDAR-based methods, which are enhanced by adding RGB data and instance segmentation data. We also selected several recent monocular image-based 3D detection methods for comparison, most of which are based on RGB image features, because of the small number of current pseudo-LiDAR-based methods. Owing to the lack of spatial features, all other methods performed worse than the pseudo-LiDAR-based methods. Figure 4 shows the results of the 3D bounding box estimation on the KITTI validation set, where the red bounding box represents the ground truth data, the blue bounding box indicates the result of our estimation, and the green bounding box represents the estimation result of MonoDETR. The first three rows show the results of projecting the 3D bounding box onto the 2D image, and the final row shows the results of projecting the 3D bounding box onto the pseudo-LiDAR point cloud. The pseudo-LiDAR point cloud was generated using the proposed method. It is clear from the visualized data that the 3D bounding boxes estimated by both our proposed method and MonoDETR overlapped significantly with the ground truth 3D bounding box. However, MonoDETR could not detect occluded objects as well as distant objects, our proposed methods performed better than MonoDETR.

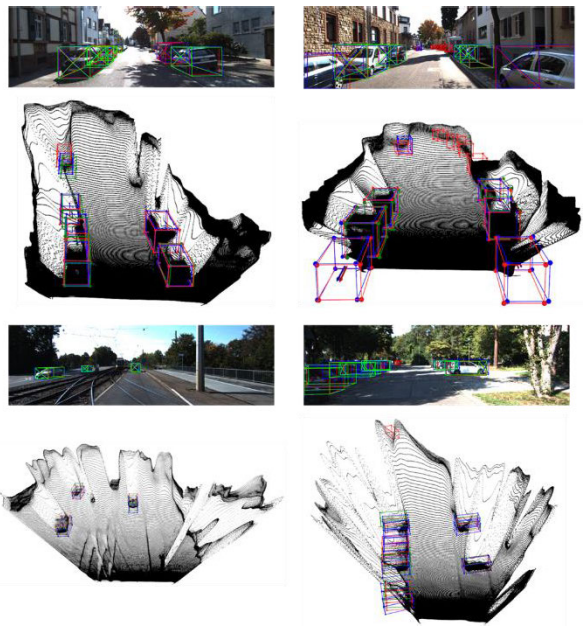

FIGURE 4. Qualitative results of our method and MonoDETR on KITTI validation set.

E. ABLATION STUDY

In this section, we describe the analysis of the effect of the 2D object detection results on the 3D object detection performance, the effect of depth estimation networks on the 3D object detection performance, and the effect of 3D object detection networks on the performance.

The pseudo-LiDAR point clouds containing the instance mask and bounding box are depicted in Figure 5. The red

TABLE 5. Performance comparison of different 2D guided data.

Method	APBEV/AP3D (%), IoU = 0.5			APBEV/AP3D (%), IoU = 0.7		
	Easy	Moderate	Hard	Easy	Moderate	Hard
No 2D info	70.3/65.9	49.2/45.1	42.5/39.1	35.8/25.8	22.7/16.1	20.8/15.4
Bounding box	74.0/69.8	55.1/50.8	48.1/44.9	48.2/32.8	33.5/23.4	28.2/20.0
Instance mask	**75.8/71.1**	**56.2/52.1**	**52.0/47.1**	**44.7/34.7**	**31.8/25.0**	**27.9/21.2**

TABLE 6. Performance comparison when using different depth estimation networks.

Method	Depth estimation network	APBEV/AP3D (%), IoU = 0.5			APBEV/AP3D (%), IoU = 0.7		
		Easy	Moderate	Hard	Easy	Moderate	Hard
Pseudo-LiDAR	DORN	70.8/66.3	49.4/42.3	42.7/38.5	40.6/28.2	26.3/18.5	22.9/16.4
Ours	DORN	72.6/65.9	55.0/50.1	48.2/44.6	42.5/29.1	29.5/20.4	26.3/17.7
	Depth Anything	**75.8/71.1**	**56.2/52.1**	**52.0/47.1**	**44.7/34.7**	**31.8/25.0**	**27.9/21.2**

TABLE 7. Performance comparison with and without self-attention module.

Method	APBEV/AP3D (%), IoU = 0.5			APBEV/AP3D (%), IoU = 0.7		
	Easy	Moderate	Hard	Easy	Moderate	Hard
PointPillars	70.3/65.9	49.2/45.1	42.5/39.1	35.8/25.8	22.7/16.1	20.8/15.4
Proposed method	**75.8/71.1**	**56.2/52.1**	**52.0/47.1**	**44.7/34.7**	**31.8/25.0**	**27.9/21.2**

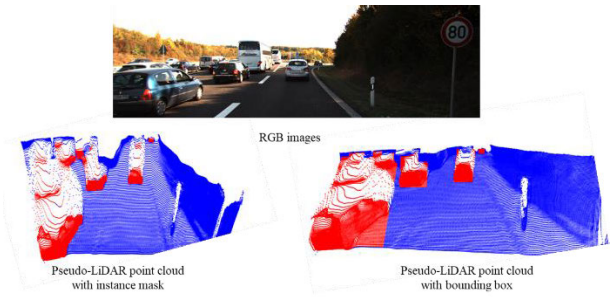

FIGURE 5. Pseudo-LiDAR point clouds with instance mask and bounding box.

part of the figure represents the point cloud containing 2D information, which is the point cloud region with non-zero confidence, whereas the blue part is the normal point cloud, which is the point cloud region with zero confidence. It can be observed that the point cloud containing the confidence level based on the instance mask was more accurate for the object part. The point cloud containing the bounding box contained more background point cloud information owing to the many background pixels in the bounding box.

It is evident from Table 5 that 3D detection can be guided effectively by adding 2D detection information. The depth maps were all predicted by Depth Anything, and the 3D detection networks used were all PointPillars-based self-attention 3D detection networks, as described in Section III. The difference lies in the confidence level of the point cloud data. "No 2D info" indicates that all confidence levels were zero, and neither 2D detection results were used as bootstrap data. "Bounding box" indicates that the 2D bounding box of the object extracted by YOLOv8 and its score were used as the confidence level of the point cloud. "Instance mask" indicates that the semantic segmentation of the object extracted by Mask2Former [40] was used as the confidence level of the point cloud mask and its score was used as the confidence level of the point cloud. In the detection of "hard" level objects with the threshold set to 0.5, the method using 2D detection masks improved the average BEV accuracy by 9.5% compared to the method without 2D detection information. This result demonstrates the significant role of the 2D detection mask in enhancing 3D detection networks for long-range object detection. In summary, the 3D prediction performance can be effectively improved by including 2D object detection information.

Monocular depth estimation is a fundamental problem in fields such as autonomous driving, robotics, and virtual reality. With the development of convolutional networks, a series of deep-learning-based architectures have emerged [36], [37], [53], which effectively improve the performance of monocular depth estimation and provide the basis for our pseudo-LiDAR methodology. Because the performance of our proposed method is determined by the monocular depth estimation, we compared the 3D detection performance when different monocular depth networks were used.

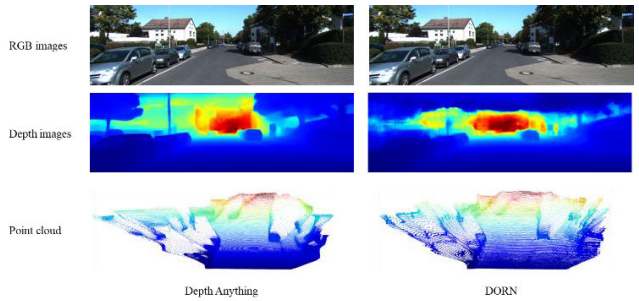

FIGURE 6. Depth maps and point clouds for different monocular depth estimation networks.

Figure 6 shows the depth results obtained from the different depth estimation networks and the converted pseudo-LiDAR point clouds. It can be observed that the gap between the converted point clouds was very large owing to the varying depth maps obtained by the different depth estimation networks. Therefore, depth estimation results can have a significant effect on 3D object detection networks.

The accuracy results obtained by our algorithm based on two different depth estimation networks, namely DORN [37] and Depth Anything [36], are listed in Table 6. The impact on the 3D object detection performance was significant owing to the difference in the depth estimation performance. Accurate depth estimation can effectively improve the 3D detection performance. In addition, our proposed method outperformed pseudo-LiDAR using the DORN depth estimation network. This also demonstrates that the success of the proposed method is not always based on the improved performance of the depth estimation network.

We compared PointPillars with a network that included added self-attention to verify the effectiveness of the self-attention module. This experiment was also based on the network structure presented in Section III, and only the 3D detection network was different. It can be observed from Table 7 that the 3D detection network with the self-attention module significantly outperformed the PointPillars network without self-attention. This demonstrates the effectiveness of adding the self-attention module.

V. CONCLUSION

We have proposed a monocular 3D object detection system that is distinguished from previous methods in its use of pseudo-LiDAR point clouds as the foundation for 3D object detection. This innovative approach leverages 2D detection results as guidance to generate point cloud data, and the novel 3D detection network that is specifically tailored for pseudo-LiDAR point clouds further improves the detection accuracy and robustness. In addition, a zero-shot-based depth prediction model is used to improve the depth prediction accuracy and overall generalization capability of the system. This model allows for the processing of any input without requiring pre-training, thereby increasing its adaptability. The depth estimation architecture effectively enhances the

TABLE 8. Notation table.

Symbol	Definition
(u, v)	The 2D position of each pixel in the image, where u represents the horizontal (column) and v represents the vertical (row) index.
(x, y, z)	The position of a point in 3D space, where x represents the horizontal direction, y represents the vertical direction, and z represents the depth (front to back direction).
(h, w, l)	h represents the height, w represents the width, and l represents the length of the object.
θ	The heading angle θ describes the orientation of the object in 3D space.
(C_x, C_y)	The optical center (or principal point) on the image plane, which is the point where the optical axis of the camera intersects with the image plane.
f	The focal length of a camera, which represents the distance from the optical center of the camera (the center of the lens) to the image plane.
M_E	A 4 × 4 matrix that represents the extrinsic parameters of a camera, which transform points from the world coordinate system into the camera coordinate system.
R	A 3 × 3 orthogonal rotation matrix that describes the orientation of the camera.
t	A 3D translation vector (3 × 1 column vector) that represents the position or displacement of the camera in the world coordinate system.
$\begin{bmatrix} x \\ y \\ z \end{bmatrix}$	A point in the world coordinate system.
M_E^{-1}	The inverse of the extrinsic matrix, which transforms a point from the camera coordinate system back into the world coordinate system.
$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$	A 3D point in the camera coordinate system (i.e., the coordinates of a point as seen by the camera).
p_c	The pseudo-LiDAR point cloud.
$conf$	The confidence of each pixel of the predicted 2D bounding box.
$(x, y, z, conf, xc, yc, zc, xp, yp, w_{height}, w_{2d})$	xc, yc, zc denote the distances to the arithmetic mean points of all points in the grid; $conf$ denotes the confidence of each pixel of the predicted 2D bounding box; xp, yp denote the offsets of the points to the x, y center of the grid, respectively; and w_{height}, w_{2d} denote the weights of the height range and 2D mask, respectively.
(D, P, N)	P represents the maximum number of non-empty pillars (pillar grids containing points) per sample. N is the maximum number of points within each pillar. D refers to the number of features or dimensions for each point.
(C, H, W)	C represents the number of channels in the pseudo-image; H refers to the height of the pseudo-image, representing the number of vertical grid cells in the BEV projection of the point cloud; and W is the width of the pseudo-image, representing the number of horizontal grid cells in the BEV projection.
$Conv2D$ $(C_{in}, C_{out}, k, s, p)$	$Conv2D(C_{in}, C_{out}, k, s, p)$ represents a 2D convolution operator, where C_{in} and C_{out} are the numbers of input and output channels, respectively; and k, s, p are the kernel size, step size, and padding size, respectively.

TABLE 8. ***(Continued.)*** **Notation table.**

$Q \in \mathbb{R}^{C_Q \times N}$, $K \in \mathbb{R}^{C_K \times N}$, $V \in \mathbb{R}^{C_V \times N}$	Q (query), K (key), and V (value) are vectors that originate from the encoded point cloud features. C_Q , C_K , and C_V represent the numbers of channels for the query, key, and value vectors, respectively. N is the number of spatial positions.
$O_i = \sum_{j=1}^{N} \omega_{i,j} \psi(x_j)$	O_i is the output context at position i and $\omega_{i,j}$ is the normalized attention weight. ψ indicates a fully connected layer that transforms x (the flattened feature map) into value space.
$W = softmax(\frac{Q^T \times K}{\sqrt{C_K}})$	Q^T is the transpose of the query matrix Q of shape $N \times C_Q$, resulting in a matrix of shape $N \times C_Q$; K is the key matrix of shape $C_K \times N$; $\sqrt{C_K}$ is a scaling factor; and $softmax$ is applied to the scaled dot product.
$(x, y, z, w, l, h, \theta)$	w is the width of the bounding box along the x-axis; l is the length of the bounding box along the y-axis; h is the height of the bounding box along the z-axis; and θ is the orientation angle of the bounding box, which defines how the box is rotated around its center, typically around the vertical axis.
$\Delta x = \frac{x^{gt} - x^{pred}}{d^{pred}}$, $\Delta y = \frac{y^{gt} - y^{pred}}{d^{pred}}$, $\Delta z = \frac{z^{gt} - z^{pred}}{h^{pred}}$, $\Delta w = log \frac{w^{gt}}{w^{pred}}$, $\Delta l = log \frac{l^{gt}}{l^{pred}}$, $\Delta h = log \frac{h^{gt}}{h^{pred}}$, $\Delta\theta = sin(\theta^{gt} - \theta^{pred})$	Δx is the normalized difference in the x-coordinates of the ground truth x^{gt} and predicted x^{pred} bounding boxes, scaled by d^{pred}; Δy is the normalized difference in the y-coordinates; Δz is the normalized difference in the z-coordinates; Δw is the logarithmic difference in width; Δl is the logarithmic difference in length; Δh is the logarithmic difference in height; and $\Delta\theta$ is the sine of the difference in orientation angles.
$\mathcal{L}_{loc} = \sum_{b \in (x,y,z,w,l,h,\theta)} SmoothL1(\Delta b)$	Δb represents the difference for each bounding box parameter b , as defined previously. $SmoothL1(\Delta b)$ is the Smooth L1 loss, which is a loss function that combines properties of both the L1 and L2 losses.
$\mathcal{L}_{cls} = -\alpha_{pred}(1 - p^{pred})^{\gamma} log p^{pred}$	p^{pred} denotes the probability of the predicted object class. α_{pred} is a weighting factor that balances the importance of positive/negative classes. γ is a focusing parameter that adjusts the rate.

accuracy compared to previous depth estimation networks. Specifically, the results of the ablation experiments indicated that this approach yielded an average accuracy increase of 3.3% when the threshold was set to 0.7. In addition, the bounding boxes obtained from 2D detection are integrated as confidence data into the pseudo-LiDAR point cloud, thereby improving the 3D detection performance through the guidance provided by the 2D detection masks. The experimental results demonstrated that, when the detection threshold was set to 0.5 for ''hard'' objects, the method that incorporates 2D detection masks achieved an average BEV accuracy improvement of 9.5% compared to methods that do not use these masks. Given the continuous advancements in monocular depth estimation and 2D object detection algorithms, we designed both modules to function as independent offline components. This design allows for the seamless integration of the latest algorithms to ensure that the framework maintains high flexibility and state-of-the-art performance. Finally, this research includes the redesign of a novel 3D detection network. We developed a novel adaptive weight pillar feature encoder, which retains more foreground points while mitigating the impact of excessive background points on 3D detection, to address the challenges posed by dense pseudo-LiDAR data. Following this, we designed a new feature extraction network that combines the global features extracted by a Transformer with the local features obtained by a CNN. The experiments showed that this hybrid approach improved the 3D detection accuracy, achieving a 7.3% increase in the average BEV precision at a threshold of 0.5 compared to previous networks such as PointPillars, which are based on LiDAR.

In summary, the effectiveness of the proposed method was validated through experiments conducted on the KITTI dataset, demonstrating significant improvements over current state-of-the-art monocular 3D object detection networks. However, owing to the limitations of the KITTI dataset, the proposed method has yet to be evaluated in other complex environments, such as low-light conditions or rainy weather. In future research, we plan to investigate and analyze the performance in these challenging scenarios and explore lightweight system designs that are suitable for applications in autonomous driving and robotic navigation.

APPENDIX
NOTATIONS

Table 8 provides clear definitions for all variables and symbols used in our study.

REFERENCES

[1] X. Ma, W. Ouyang, A. Simonelli, and E. Ricci, ''3D object detection from images for autonomous driving: A survey,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 5, pp. 3537–3556, May 2024.

[2] R. Gao, J. Park, X. Hu, S. Yang, and K. Cho, ''Reflective noise filtering of large-scale point cloud using multi-position LiDAR sensing data,'' *Remote Sens.*, vol. 13, no. 16, p. 3058, Aug. 2021.

[3] R. Gao, M. Li, S.-J. Yang, and K. Cho, ''Reflective noise filtering of large-scale point cloud using transformer,'' *Remote Sens.*, vol. 14, no. 3, p. 577, Jan. 2022.

[4] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teulière, and T. Chateau, ''Deep MANTA: A coarse-to-fine many-task network for joint 2D and 3D vehicle analysis from monocular image,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1827–1836.

[5] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, ''Monocular 3D object detection for autonomous driving,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2147–2156.

[6] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, ''3D object proposals for accurate object class detection,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, Dec. 2015, pp. 424–432.

[7] W. Ouyang and X. Wang, ''Joint deep learning for pedestrian detection,'' in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2056–2063.

[8] X. Ma, Z. Wang, H. Li, P. Zhang, W. Ouyang, and X. Fan, ''Accurate monocular 3D object detection via color-embedded 3D reconstruction for autonomous driving,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6850–6859.

[9] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-LiDAR from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8437–8445.
[10] W. Chen, Y. Li, Z. Tian, and F. Zhang, "2D and 3D object detection algorithms from images: A survey," *Array*, vol. 19, Sep. 2023, Art. no. 100305.
[11] S. Y. Alaba and J. E. Ball, "Deep learning-based image 3-D object detection for autonomous driving: Review," *IEEE Sensors J.*, vol. 23, no. 4, pp. 3378–3394, Feb. 2023.
[12] X. Ma, S. Liu, Z. Xia, H. Zhang, X. Zeng, and W. Ouyang, "Rethinking pseudo-LiDAR representation," in *Proc. 16th Eur. Conf. Comput. Vis. (ECCV)*, Glasgow, U.K. Cham, Switzerland: Springer, Aug. 2020, pp. 311–327.
[13] J. Li, C. Luo, and X. Yang, "PillarNeXt: Rethinking network designs for 3D object detection in LiDAR point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 17567–17576.
[14] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 770–779.
[15] C. R. Qi, Y. Li, H. Su, and L. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. neural Inf. Process. Syst.*, vol. 30, Jan. 2017, pp. 1–10.
[16] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4490–4499.
[17] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, Oct. 2018.
[18] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12689–12697.
[19] Z. Liu, H. Tang, S. Zhao, K. Shao, and S. Han, "PVNAS: 3D neural architecture search with point-voxel convolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8552–8568, Nov. 2022.
[20] S. Shi, L. Jiang, J. Deng, Z. Wang, C. Guo, J. Shi, X. Wang, and H. Li, "PV-RCNN++: Point-voxel feature set abstraction with local vector representation for 3D object detection," *Int. J. Comput. Vis.*, vol. 131, no. 2, pp. 531–551, Feb. 2023.
[21] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, Dec. 2015, pp. 91–99.
[22] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," 2019, *arXiv:1904.01355*.
[23] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*.
[24] A. Mousavian, D. Anguelov, J. Flynn, and J. Košecká, "3D bounding box estimation using deep learning and geometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5632–5640.
[25] X. Ma, Y. Zhang, D. Xu, D. Zhou, S. Yi, H. Li, and W. Ouyang, "Delving into localization errors for monocular 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4719–4728.
[26] T. Wang, X. Zhu, J. Pang, and D. Lin, "Probabilistic and geometric depth: Detecting objects in perspective," in *Proc. Conf. Robot Learn.*, Jun. 2021, pp. 1475–1485.
[27] A. Mallya and S. Lazebnik, "PackNet: Adding multiple tasks to a single network by iterative pruning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7765–7773.
[28] R. Zhang, H. Qiu, T. Wang, Z. Guo, Z. Cui, Y. Qiao, H. Li, and P. Gao, "MonoDETR: Depth-guided transformer for monocular 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 9121–9132.
[29] X. Wang, W. Yin, T. Kong, Y. Jiang, L. Li, and C. Shen, "Task-aware monocular depth estimation for 3D object detection," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 7, pp. 12257–12264.
[30] X. Ye, L. Du, Y. Shi, Y. Li, X. Tan, J. Feng, E. Ding, and S. Wen, "Monocular 3D object detection via feature domain adaptation," in *Proc. 16th Eur. Conf. Comput. Vis. (ECCV)*, Glasgow, U.K. Cham, Switzerland: Springer, Aug. 2020, pp. 17–34.
[31] R. Qian, D. Garg, Y. Wang, Y. You, S. Belongie, B. Hariharan, M. Campbell, K. Q. Weinberger, and W.-L. Chao, "End-to-end pseudo-LiDAR for image-based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5880–5889.
[32] Y. Wang, B. Yang, R. Hu, M. Liang, and R. Urtasun, "PLUMENet: Efficient 3D object detection from stereo images," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 3383–3390.
[33] X. Weng and K. Kitani, "Monocular 3D object detection with pseudo-LiDAR point cloud," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 857–866.
[34] R. Gao, J. Xu, Y. Chen, and K. Cho, "Heterogeneous feature fusion module based on CNN and transformer for multiview stereo reconstruction," *Mathematics*, vol. 11, no. 1, p. 112, Dec. 2022.
[35] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, "TransUNet: Transformers make strong encoders for medical image segmentation," 2021, *arXiv:2102.04306*.
[36] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth anything: Unleashing the power of large-scale unlabeled data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 10371–10381.
[37] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2002–2011.
[38] G. Jocher, A. Chaurasia, and J. Qiu. *Ultralytics YOLO V8*. Accessed: Dec. 11, 2024. [Online]. Available: https://github.com/ultralytics/ultralytics
[39] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D object detection from RGB-D data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 918–927.
[40] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1280–1289.
[41] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 784–799.
[42] Y. Wu, Y. Chen, L. Yuan, Z. Liu, L. Wang, H. Li, and Y. Fu, "Rethinking classification and localization for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10183–10192.
[43] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.
[44] G. Song, Y. Liu, and X. Wang, "Revisiting the sibling head in object detector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11560–11569.
[45] H. Meng, C. Li, G. Chen, L. Chen, and A. Knoll, "Efficient 3D object detection based on pseudo-LiDAR representation," *IEEE Trans. Intell. Vehicles*, vol. 9, no. 1, pp. 1953–1964, Jan. 2024.
[46] A. Geiger and U. R. Lenzp, "Are we ready for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.
[47] X. Chu, J. Deng, Y. Li, Z. Yuan, Y. Zhang, J. Ji, and Y. Zhang, "Neighbor-vote: Improving monocular 3D object detection through neighbor distance voting," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 5239–5247.
[48] J. Ansel, E. Yang, H. He, N. Gimelshein, A. Jain, M. Voznesensky, B. Bao, P. Bell, D. Berard, E. Burovski, and G. Chauhan, "PyTorch 2: Faster machine learning through dynamic Python bytecode transformation and graph compilation," in *Proc. 29th ACM Int. Conf. Architectural Support Program. Lang. Operating Syst.*, vol. 2, Apr. 2024, pp. 929–947.
[49] OpenPCDet Development Team. *OpenPCDet: An Open-Source Toolbox for 3D Object Detection From Point Clouds*. Accessed: Dec. 11, 2024. [Online]. Available: https://github.com/open-mmlab/OpenPCDet
[50] K. Huang, T. Wu, H. Su, and W. H. Hsu, "MonoDTR: Monocular 3D object detection with depth-aware transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4002–4011.
[51] Z. Min, B. Zhuang, S. Schulter, B. Liu, E. Dunn, and M. Chandraker, "NeurOCS: Neural NOCS supervision for monocular 3D object localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 21404–21414.
[52] L. Yan, P. Yan, S. Xiong, X. Xiang, and Y. Tan, "MonoCD: Monocular 3D object detection with complementary depths," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 10248–10257.
[53] J. Kim, R. Gao, J. Park, J. Yoon, and K. Cho, "Switchable-encoder-based self-supervised learning framework for monocular depth and pose estimation," *Remote Sens.*, vol. 15, no. 24, p. 5739, Dec. 2023.

RUI GAO received the B.S. degree in computer science and technology from Linyi University, Linyi, China, in 2017, and the M.Eng. degree in multimedia engineering from Dongguk University, Seoul, South Korea, in 2020, where he is currently pursuing the Ph.D. degree with the Department of Multimedia Engineering. His current research interests include autonomous, monocular 3-D detection, artificial intelligence, and deep learning.

JUNOH KIM received the Ph.D. degree in multimedia engineering from Dongguk University, Seoul, South Korea, in 2024. He has accumulated over 24 years of experience working in technology at computer language compiler and IoT companies. Currently, he has been a Professor with Dongguk University, where he participates in projects related to game AI, autonomous vehicles, and robotics, 3-D reconstruction, since 2016. His research interests include reinforcement learning for robots and ships and monocular-based 3-D reconstruction.

KYUNGEUN CHO (Member, IEEE) received the B.Eng. degree in computer science, and the M.Eng. and Dr.-Eng. degrees in computer engineering from Dongguk University, Seoul, South Korea, in 1993, 1995, and 2001, respectively. From 1997 to 1998, she was a Research Assistant with the Institute for Social Medicine, Regensburg University, Germany; and a Visiting Researcher with the FORWISS Institute, TU-Munchen University, Germany. She has been a Full Professor with the Department of Multimedia Engineering, Dongguk University, since September 2003. She has led several projects on robotics and game engines. She has published many technical articles in her research areas. Her current research interests include artificial intelligence of robots and virtual characters and real-time computer graphics technologies.